

# **METHOD FOR GENERATING CONSTRAINED COMPONENT PLACEMENT FOR INTEGRATED CIRCUITS AND PACKAGES**

## **CROSS REFERENCE TO RELATED APPLICATIONS**

**[0001]** This application claims priority from United States Provisional Patent Application No. 60/415,625, filed October 2, 2002.

## **BACKGROUND OF THE INVENTION**

**[0002]** 1. **Field of the Invention**

**[0003]** The present invention relates to circuit design and, more particularly, to component placement in a circuit design.

**[0004]** 2. **Description of the Prior Art**

**[0005]** Integrated circuit and electronic board/package design is subject to complex geometric and electrical rules governing the allowed placement of components. Component placements which fail to abide by these rules may occupy too much area to be economically manufactured, achieve levels of desired performance far below those specified at the outset of the design process, or even fail to perform their desired function correctly.

**[0006]** A “component” may be an individual electronic device (for example, a transistor) or a set of pre-placed devices to be treated as a single, atomic placeable object. A “group” may be a hierarchy of placed components, such as several devices of a sub-circuit, or several sub-circuits placed in a circuit to also be treated as single, atomic object.

**[0007]** A component is conventionally specified by a rectilinear, typically rectangular, outline which completely encompasses its internal geometry and any extra space that must be included around the exterior of the component to ensure compliance with manufacturing rules that specify allowable distances between components. The result of geometrically arranging a plurality of such components on the surface of an IC chip or a board is called a “placement”, or a “layout”, or a “floorplan”.

**[0008]** It is conventional to optimize component placements to minimize some combination of the overall area of a layout, the length of conducting wires (also called “nets”) that connect the components, and the delay to propagate an electrical signal along

selected paths of the longest wires in the design. For placement of digital components, strategies of this type often suffice. However, for placements intended to create layouts that incorporate non-digital components, for example, analog components, mixtures of analog and digital components (called “mixed-signal” designs), radio-frequency components (“RF”) and microwave-frequency components, among others, additional geometric constraints are often imposed. These additional geometric constraints maximize the precision with which various electrical quantities (current, voltage, charge, capacitance, inductance, etc.) may be created or manipulated by these placed components, thus enhancing the likelihood of correct electrical function of the final manufactured design.

[0009] These geometric constraints come in several well-known styles. The geometric placement constraint styles of relevance to the present invention include symmetries, groups and isolation structures.

#### Symmetries:

[0010] Numerous prior art techniques exist for generating a layout of components when no symmetry, grouping, and/or isolation constraints are present. These conventionally comprise a method to represent the location of each placed component and a method to iteratively perturb or amend this representation to incrementally improve it to a best-quality placement, as measured by a cost function that quantifies the goodness of the placement.

[0011] The present invention is a new method for placement that supports standard prior art placement goals for digital components, e.g., wirelength, area, etc., and also supports symmetry, grouping and isolation requirements often associated with analog components.

[0012] With reference to Fig. 1, a placement problem typically comprises a set of arrangeable components that may each (for simplicity of description) be treated as rectangular objects. These rectangular objects must be arranged on a layout surface which itself can be treated as a rectangle.

[0013] Individual components carry geometric rules that determine the minimum allowed proximity (distance) to each other individual component. In particular, it is

normal that placed components may not overlap. The process of “placement” assigns to each rectangular component an absolute (x, y) location, indexed relative to x (horizontal) and y (vertical) coordinate axes suitably defined for the layout surface itself.

**[0014]** With reference to Fig. 2, symmetry constraints impose mirror-like placement constraints on pairs of components. The two elements in the pair are referred to as siblings. A distinguished line segment - called a symmetry line - defines an axis of symmetry for the constraint.

**[0015]** In the simplest form of symmetry, a pair-symmetric constraint mandates that a pair of identical components must be placed as “mirror reflections” of each other about the symmetry line. For a vertical symmetry line, this means identical y (vertical) coordinates, but x coordinates that locate each component an identical distance D away from the symmetry line.

**[0016]** With reference to Fig. 3, and with continuing reference to Fig. 2, the symmetry line can be horizontal or vertical. To this end, the roles of the x and y coordinate values can be interchanged. A distinguished line segment again defines an axis of symmetry for the constraint, only now, this symmetry line is horizontal with respect to the layout surface. For a horizontal symmetry line, this means identical x (horizontal) coordinates, but y coordinates that locate each component an identical distance D away from the central symmetry line.

**[0017]** With reference to Fig. 4, in addition, the pair of components may be required to be in a specific relative orientation with respect to the symmetry line. This can apply for either the vertical or horizontal cases. Any geometric object being placed on a two-dimensional layout surface can be rotated or reflected into one of the eight unique orientations shown in Fig. 5. Pair-symmetric constraints can also optionally mandate the orientation of each component of the pair.

**[0018]** In addition, the pair of components may be required to be in a specific relative orientation with respect to the symmetry line. This can apply for either the vertical or horizontal cases. As shown in Fig. 6, simple pair-symmetric orientation requires the components to be arranged in identical orientations on each side of the line. (Identical left-right orientation for vertical symmetry lines or identical top-bottom orientation for horizontal symmetry lines). However, as shown in Fig. 7, mirror pair-symmetric

orientation requires the components to be arranged as mirror images with respect to the symmetry line. (Left-right reversal across a vertical symmetry line or top-bottom reversal across a horizontal symmetry line).

**[0019]** Another type of symmetry constraint, shown in Figs. 8 and 9, is a self-symmetry constraint. This requires a single component to have its center coordinate located on the symmetry line. Thus, the self-symmetric component extends a distance  $D$  on each side of the symmetry line. As shown in Figs. 8 and 9, self-symmetry can be applied to either a vertical symmetry line or a horizontal symmetry line. As with the case of the pair-symmetric constraint, an optional orientation constraint may be imposed on the single self-symmetric component.

**[0020]** With reference to Figs. 10 and 11, another type of symmetry constraint is an asymmetry constraint. This requires a single component to have an internal coordinate other than its center point located on the symmetry line. Thus, the asymmetric component extends a distance  $D1$  on one side of the symmetry line and a distance  $D2$  on the other side of the symmetry line, where  $D1+D2$  equals the dimension of the component perpendicular to the symmetry line.

**[0021]** As shown in Figs. 10 and 11, asymmetry can be applied to either a vertical or a horizontal symmetry line. As with the case of the pair-symmetric constraint, an optional orientation constraint may be imposed on the single asymmetric component.

**[0022]** With reference to Figs. 12 and 13, another type of symmetry constraint is a fixed symmetric offset constraint. This requires a single component to not overlap the symmetry line but, instead, be separated from the symmetry line by a precise (“fixed”) distance  $D$ . Thus, the fixed symmetric offset component is exactly a distance  $D$  away on one side of the symmetry line. Another required element of the fixed symmetric offset constraint is which side of the symmetry line the component is required to reside in. As shown in Figs. 12 and 13, fixed symmetric offset constraints can appear with either vertical or horizontal symmetry lines. As with the case of other symmetry constraints, an optional orientation constraint may be imposed on the component.

**[0023]** With reference to Fig. 14, components that do not participate in any symmetry constraints are referred to as non-symmetric. Symmetric and non-symmetric components can co-exist in the same placement on the same layout surface. As shown in Fig. 15,

multiple different symmetry lines can also co-exist on a single layout surface. Individual components will be identified as either non-symmetric or bound to a particular symmetry line. When the symmetry line extends across the entire layout surface, it is referred to as a global symmetry line. When the symmetry line extends only across a portion of the layout surface, it is referred to as a local symmetry line. Multiple local and global, horizontal and vertical symmetry lines can arbitrarily co-exist on the same layout surface, each with an arbitrary collection of components with arbitrary symmetry constraints imposed on them.

#### Groups:

**[0024]** With reference to Figs. 16 and 17, group constraints require sets of individual components to be placed near each other. In addition, group constraints define, for the set of components in a group, a minimum bounding rectangle. This rectangle defines the components of the group, and imposes the additional constraint that no component that is not part of the group can be placed inside this rectangle.

**[0025]** With reference to Fig. 18, the minimum bounding rectangle of a group can be treated like a single placeable component. Thus, groups can also have symmetry constraints. A pair of identical groups can form a symmetric pair. A single group can be self-symmetric, asymmetric, non-symmetric, or can have a fixed symmetric offset. Groups can be bound to local or global symmetry lines, and horizontal or vertical symmetry lines.

**[0026]** With reference to Fig. 19, the minimum bounding rectangle of a group can also or alternatively be treated like it was the layout surface itself. Thus groups can also have internal, local symmetry lines. Components inside of a group can be bound to this local symmetry line or they can be non-symmetric. The local symmetry line can be group-global, namely, it traverses the entire dimension of the group, or it can be group-local, namely, it traverses partially but not entirely across one dimension of the group. A single group can have any number of such internal symmetry lines. A group with internal symmetries can itself be treated as a single minimum bounding rectangle that obeys any symmetry constraint. These are referred to as hierarchical symmetry constraints.

### Isolation Structures:

**[0027]** With reference to Figs. 20 and 21, isolation constraints handle situations where individual or hierarchical placed components or groups may be electrically sensitive to disruption from other components or groups placed nearby. One mechanism for mitigating this problem is to add a special, isolation structure around portions of the perimeter of each sensitive component. These isolation structures appear as geometric bands around portions or all of the perimeter of a component, increasing its size, and changing its spacing rules with respect to other placed components or groups.

**[0028]** With reference to Figs. 22 and 23, groups can also have isolation constraints. For groups, these isolation structures appear as geometric bands around portions or all of the perimeter of the group's minimum bounding rectangle, changing its geometric size and spacing rules with respect to other placed components or groups.

**[0029]** All of these three styles of constraints, namely, symmetries, groups and isolation structures, can co-exist in one placement. Individual components may participate in symmetry constraints, be parts of groups, or have individual isolation structures. Likewise, groups themselves can have internal symmetries, participate in group/hierarchical symmetries external to themselves, and have group-level isolation structures. Groups can be nested inside each other to define arbitrary layout hierarchies.

**[0030]** Many prior art methods exist for generating a layout of components when symmetry, grouping, and isolation constraints are not needed. These conventionally comprise a method to represent the location of each placed component, and a method to iteratively perturb this representation to incrementally improve it to a best-quality placement, as measured by an auxiliary cost function.

**[0031]** These prior art methods, however, do not work when the presence of arbitrary combinations of symmetry, grouping, or isolation constraints exist. While prior art methods exist that partially address some of these constraints, all of the prior art methods for handling symmetries suffer from at least one of the following problems:

- They cannot guarantee the ultimate geometric legality of the placement, e.g., components may overlap;
- They cannot guarantee that all symmetry constraints are satisfied in the final placement;

- The process of evaluating the feasibility of satisfying all symmetries for each placement solution candidate is inefficient, e.g., quadratic in the number of components;
- They cannot support the full range of required constraints required, namely, self and pair symmetry, asymmetry and non-symmetry, fixed symmetric offsets, groups, group and hierarchical symmetries, isolation structures for components and groups.

**[0032]** The present invention overcomes the above problems and others by providing a method that utilizes a novel data structure called a tree structure that supports all of the foregoing constraints. Every placement solution candidate generated during iterative improvement will satisfy all the constraints. The method of generating these solution candidates, and the method of evaluating the cost function which determines the quality of each solution candidate, are both novel and efficient. The method requires only a single binary tree data structure with suitable annotations. Evaluating the quality of any placement solution requires only searches of this tree structure. In particular, the method does not require the construction of any secondary or auxiliary data structures to check that the constraints can be met, or to evaluate the cost of a placement.

### SUMMARY OF THE INVENTION

**[0033]** The invention is a computer-implemented method for determining component placement in a circuit formed on a substrate, such as a printed circuit board or a semiconductor, i.e., an integrated circuit. The method includes (a) receiving a plurality of components, with each component having associated therewith a width, a height and one of a symmetric and a non-symmetric placement constraint and (b) creating a tree structure that expresses the placement constraints for the plurality of components. The tree structure includes a global root node that represents a global symmetry line, a leaf representing each component, with the component associated with each leaf tagged for placement on a first side, a second side, or on both sides of the global symmetry line, and at least one interior node that represents a slicing line that establishes a relative placement of at least two components with respect to each other on the same side of the global symmetry line. The method further includes (c) performing at least one structured search of the tree structure to determine the initial placement of components on the first side, the second side or both sides of the global symmetry line based on the tagging of each

component and the connections of the leaves, the global node and the interior nodes to form the tree structure; and (d) performing another structured search of the tree structure to determine the final placement of components based on at least one of the widths and heights of the components.

**[0034]** Each structured search can be a depth-first search. Each slicing line can have a direction that is parallel or perpendicular to the global symmetry line and step (c) can include the placement of said at least two components with respect to each other as a function of the direction of the slicing line.

**[0035]** Step (c) can include, for components tagged for placement on the same side of the global symmetry line, placing a component associated with a leaf positioned lower in the tree structure closer to the global symmetry line than a component associated with a leaf positioned higher in the tree structure. Each node can have a pair of branches descending therefrom and each branch can couple the node from which it descends to (1) one of the interior nodes, (2) one of the leaves, or (3) a null.

**[0036]** Step (c) can include, for an interior node that represents a slicing line that is parallel to the global symmetry line, placing on one side of the global symmetry line, a first component which is tagged for such placement and which is represented by a leaf which is connected to said interior node via one of its branches and placing on a side of the first component opposite the global symmetry line a second component which is tagged for placement on the one side of the global symmetry line and which is represented by a leaf which is connected to said interior node via the other of its branches.

**[0037]** At least one of the leaves representing the first and the second components can be connected directly to said interior node.

**[0038]** Step (c) can include, for the global root node, placing a first component which is tagged for placement on one side of the global symmetry line and which is represented by a leaf which is connected to said global root node via one of its branches and placing on a side of the first component opposite the global symmetry line, a second component which is tagged for placement on the one side of the global symmetry line and which is represented by a leaf which is connected to said global root node via the other of its branches.



**[0039]** At least one of the leaves representing the first and second components can be connected directly to the global root node.

**[0040]** Step (c) can include, for an interior node that represents a slicing line that is perpendicular to the global symmetry line, placing on one side of the global symmetry line, a first component which is tagged for such placement and which is represented by a leaf which is connected to said interior node via one branch descending therefrom and placing above or below the first component, a second component which is tagged for placement on the one side of the global symmetry line and which is represented by a leaf which is connected to said interior node via the other branch descending therefrom.

**[0041]** Step (d) can include placing one edge of a component adjacent the global symmetry line when the leaf representing said component is (1) connected to an interior node at the lowest level of the tree structure via one branch descending from said interior node or (2) connected to said interior node via the other branch descending therefrom and no other leaf is connected to said interior node via the one branch descending therefrom.

**[0042]** For each component tagged for placement on both sides of the global symmetry line in a pair-symmetric manner, step (c) includes placing first and second copies of said component on the respective first and second sides of the global symmetry line, with each copy of the component having a side closest to the global symmetry line positioned a distance D therefrom.

**[0043]** The distance D can be zero when the leaf representing the component tagged for pair-symmetric placement is connected directly to a node at the lowest level of the search tree. Alternatively, the distance D can be the greater of (1) a sum of the width(s) or height(s) of each component placed on the first side of the global symmetry line prior to placement of copies of the component tagged for pair-symmetric placement or (2) a sum of the width(s) or height(s) of each component placed on the second side of the global symmetry line prior to placement of copies of the component tagged for pair-symmetric placement.

**[0044]** For each component tagged for placement on both sides of the global symmetry line in a self-symmetric or asymmetric manner, step (c) includes placing a first part of said component on the first side of the global symmetry line and placing a second part of said component on the second side of the global symmetry line. For each component

tagged for self-symmetric placement, the first part of the component is one-half of the component and the second part of the component is the other half of the component. For each component tagged for asymmetric placement, the first part of the component is a first percentage of said component, the second part of the component is a second percentage of said component, and the sum of the first and second percentages equal one-hundred percent.

**[0045]** For each component tagged for offset symmetric placement on one side of the global symmetry line a distance  $D$  from the global symmetry line, step (c) includes placing said component on the one side of the global symmetry line with the side of said component in opposition with the global symmetry line positioned the distance  $D$  from the global symmetry line.

**[0046]** The method can further include defining an isolation structure along at least one side of at least one component and, in step (d), placing the one side of said at least one component no closer to another component or the global symmetry line than said isolation structure.

**[0047]** The method can further include amending the tree structure in at least one of the following manners: (1) changing a direction of at least one slicing line; (2) re-tagging at least one component from placement on the first side of the global symmetry line to placement on the second side of the global symmetry line, or vice versa; (3) exchanging the location of two components in the tree structure; (4) exchanging the position of two nodes in the tree structure; (5) exchanging the location of one leaf and one node, including any node and/or leaf connected to said one node; and/or (6) exchanging the width and height of a component associated with at least one leaf. Once the tree structure is amended, steps (c) and (d) can be repeated for the amended tree structure.

**[0048]** The tree structure can include a group which is tagged for placement on the first side, the second side, or both sides of the global symmetry line. The group can include a group tree structure having at least one leaf connected to a local root node that represents a local symmetry line. The component associated with each leaf of the group tree structure can be tagged for placement on a first side, a second side, or on both sides of the local symmetry line. The method can further include performing at least one structured search of the group tree structure to determine an initial placement of each component

thereof on the first side, the second side or both sides of the local symmetry line based on the tagging thereof, and performing another structured search of the group tree structure to determine a final placement of each component thereof based on at least one of the widths and the heights of the components.

**[0049]** The group tree structure can include at least one local interior node connected between the local root node and the at least one leaf. The step of performing at least one structured search of the group tree structure can include placing at least two components associated with the group tree structure with respect to each other as a function of a direction of a slicing line represented by the at least one local interior node of the group tree structure.

**[0050]** Step (c) can include performing the at least one structured search of the tree structure to determine the initial placement of the group on the first side, the second side, or both sides of the global symmetry line based on the tagging thereof, and step (d) can include performing the other structured search of the tree structure to determine the final placement of the group. Step (d) can also include substituting for the final placement of the group the final placement of the components associated therewith.

**[0051]** The local root node can be connected directly to either the root node or one of the interior nodes of the tree structure. The at least one leaf of the group tree structure can be connected directly to the local root node.

**[0052]** The invention is also a computer-implemented method for determining component placement in an integrated circuit. The method includes (a) receiving a plurality of components each having one of a symmetric and a non-symmetric placement constraint associated therewith and (b) creating a tree structure that expresses the placement constraints for the plurality of components, the tree structure including a plurality of nodes and a plurality of leaves. The plurality of nodes includes a global root node that represents a global symmetry line and a local root node connected to the global root node. The local root node represents a local symmetry line, and the local root node and at least one leaf connected thereto define a group tree structure which is tagged for placement on a first side, a second side or on both sides of the global symmetry line. Each leaf represents one of the components, with the component associated with each leaf of the group tree structure tagged for placement on a first side, a second side or on

both sides of the local symmetry line, with the component associated with each leaf of the group tree structure having at least one of a height and a width associated therewith. The method further includes (c) performing at least one structured search of the group tree structure to determine the initial placement of components thereof on the first side, the second side or both sides of the local symmetry line based on the tagging of each component and (d) performing another structured search of the group tree structure to determine the final placement of the components thereof based on at least one of the widths and heights of the components.

**[0053]** The method can further include performing at least one structured search of the tree structure to determine the initial placement of the group tree structure on the first side, the second side or both sides of the global symmetry line based on the tagging of the group tree structure, and performing another structured search of the tree structure to determine the final placement of the group tree structure.

**[0054]** The method can further include substituting the final placement of the components of a group tree structure for the final placement of the group tree structure.

**[0055]** The invention is also a computer-implemented method for determining component placement in an integrated circuit. The method includes (a) forming in the memory of the computer a tree structure that defines the placement of each of a plurality of components associated with the tree structure on a first side, a second side, or symmetrically on both sides of a symmetry line, with at least one component tagged for symmetric placement on both sides of a symmetry line; (b) performing at least one search of the tree structure to determine an initial placement of a subset of the components on the first side, the second side, or on both sides of the symmetry line; and (c) performing another search of the tree structure to determine a final placement of the subset of components, wherein at least a part of each component tagged for symmetric placement is positioned on each side of the symmetry line.

**[0056]** Each search can be a depth-first search. The tree structure can have a leaf for each component and at least one node connecting the leaves. The initial placement of the components occurs as a function of placement information, e.g., a vertical or horizontal line, associated with at least one node of the tree structure.

**[0057]** The method can further include (d) forming a new tree structure in at least one of the following manners: (1) amending the placement information associated with at least one of the nodes of the tree structure; (2) amending the tree structure to re-define the placement of at least one component from one side of the symmetry line to the other side of the symmetry line; (3) exchanging the location of two leaves in the tree structure; (4) exchanging the location of two nodes in the tree structure; (5) exchanging the location of a leaf and a node in the tree structure; and/or (6) exchanging the width and height of a component associated with a leaf of the tree structure. Steps (b) and (c) can then be repeated for the new tree structure.

**[0058]** Steps (d) and (e) can be repeated a plurality of times. For each tree structure, a cost can be determined that is related to the goodness of the final placement of the subset of components made utilizing said tree structure. The tree structure having the most favorable cost associated therewith can be selected for implementation of the final placement of the subset of components.

**[0059]** One way to determine the cost for each tree structure includes determining a ratio of (1) a total area of the subset of components in the final placement and (2) a total area of a rectangle that bounds the subset of components in the final placement.

**[0060]** The invention is also a computer readable medium having stored thereon instructions which, when executed by a processor, cause the processor to (a) receive data regarding a plurality of components, with the data for each component having associated therewith a width, a height and one of a symmetric and a non-symmetric placement constraint for the component and (b) create a tree structure that expresses the placement constraints for the plurality of components. The tree structure can include a global root node that represents a global symmetry line; a leaf representing each component, with the component associated with each leaf tagged for placement on a first side, a second side, or on both sides of the global symmetry line; and at least one interior node that represents a slicing line that establishes a relative placement of at least two components with respect to each other on the same side of the global symmetry line. The instructions can also cause the processor to (c) perform at least one structured search of the tree structure to determine the initial placement of components on the first side, the second side or both sides of the global symmetry line based on the tagging of each component and (d)

perform another structured search of the tree structure to determine the final placement of components based on at least one of the widths and heights of the components.

**[0061]** The invention is also a computer readable medium having stored thereon instructions which, when executed by a processor, cause the processor to (a) receive data regarding a plurality of components, with the data for each component including one of a symmetric and a non-symmetric placement constraint for the component and (b) create a tree structure that expresses the placement constraints for the plurality of components, the tree structure including a plurality of nodes and a plurality of leaves wherein: the plurality of nodes includes a global root node that represents a global symmetry line and at least one local root node connected to the global root node; the local root node represents a local symmetry line; a group tree structure includes the local root node and at least one leaf connected thereto; the group tree structure is tagged for placement on a first side, a second side or on both sides of the global symmetry line; and each leaf represents one of the components, with the component associated with each leaf of the group tree structure tagged for placement on a first side, a second side or on both sides of the local symmetry line, with the component associated with each leaf of the group tree structure having at least one of a height and a width associated therewith. The instructions can also cause the processor to (c) perform at least one structured search of the group tree structure to determine the initial placement of components thereof on the first side, the second side or both sides of the local symmetry line based on the tagging of each component and (d) perform another structured search of the group tree structure to determine the final placement of the components thereof based on at least one of the widths and heights of the components.

**[0062]** Lastly, the invention is a computer readable medium having stored thereon instructions which, when executed by a processor, cause the processor to (a) form in the memory of the computer a tree structure which defines the placement of each of a plurality of components associated with the tree structure on a first side, a second side, or symmetrically on both sides of a symmetry line, with at least one component tagged for symmetric placement on both sides of a symmetry line; (b) perform a search of the tree structure to determine an initial placement of a subset of the components on the first side, the second side, or on both sides of the symmetry line; and (c) perform another search of

the tree structure to determine a final placement of the subset of components, wherein at least one of a copy of or a part of each component tagged for symmetric placement is positioned on each side of the symmetry line.

**[0063]** The instructions can further cause the processor to (d) form a new tree structure in at least one of the following manners: (1) amend component placement information associated with at least one of the nodes of the tree structure; (2) amend the tree structure to re-define the placement of at least one component from the one side of the symmetry line to the other side of the symmetry line; (3) exchange the positions of two leaves in the tree structure; (4) exchange the positions of two nodes in the tree structure; (5) exchange the positions of a leaf and a node in the tree structure; and/or (6) exchange the width and height of a component associated with a leaf of the tree structure. Steps (b) and (c) can be repeated for the new tree structure.

**[0064]** The instructions can also cause the processor to repeat steps (d) and (e) a plurality of times; determine for each tree structure a cost that is related to the goodness of the final placement of the subset of components made utilizing said tree structure; and select for implementation the final placement of the subset of components made utilizing the tree structure having the most favorable cost associated therewith.

### BRIEF DESCRIPTION OF THE DRAWINGS

**[0065]** Fig. 1 shows an exemplary placement of rectangular components in a layout surface;

**[0066]** Fig. 2 shows a symmetric placement of components A1 and A2 on opposite sides of a vertical symmetry line in a layout surface;

**[0067]** Fig. 3 shows a symmetric placement of components A1 and A2 on opposite sides of a horizontal symmetry line in a layout surface;

**[0068]** Fig. 4 shows one example of a geometric orientation of a component R;

**[0069]** Fig. 5 shows eight possible geometric orientations for component R in Fig. 4;

**[0070]** Fig. 6 shows a simple pair-symmetric placement of copies of component R on opposite sides of a vertical symmetry line;

**[0071]** Fig. 7 shows a mirror image pair-symmetric placement of copies of component R on both sides of a vertical symmetry line;

[0072] Fig. 8 shows a self-symmetric placement of component R on a vertical symmetry line;

[0073] Fig. 9 shows a self-symmetric placement of component R on a horizontal symmetry line;

[0074] Fig. 10 shows an asymmetric placement of component R on a vertical symmetry line;

[0075] Fig. 11 shows an asymmetric placement of component R on a horizontal symmetry line;

[0076] Fig. 12 shows the placement of component R a fixed symmetric offset from a vertical symmetry line;

[0077] Fig. 13 shows the placement of component R a fixed symmetric offset from a horizontal symmetry line;

[0078] Fig. 14 shows a layout of symmetric and non-symmetric components in a layout surface;

[0079] Fig. 15 shows a layout of components that includes a first subset of components laid out with respect to a vertical, global symmetry line that extends across the layout surface and a second subset of components laid out with respect to a local symmetry line that extends only partially across the layout surface;

[0080] Fig. 16 shows components which define a group bounded by a minimum bounding rectangle;

[0081] Fig. 17 shows an illegal placement of a component in the group shown in Fig. 16;

[0082] Fig. 18 shows a mirror image pair-symmetric placement of copies of the group shown in Fig. 16 on opposite sides of a horizontal global symmetry line;

[0083] Fig. 19 shows a mirror image pair-symmetric placement of two groups on opposite sides of a horizontal global symmetry line, with each group including a vertical local symmetry line which is utilized for pair-symmetric placement of the components thereof;

[0084] Fig. 20 shows a placement of components, where component A is sensitive to electric disruption from component B;



[0085] Fig. 21 shows the placement of components shown in Fig. 20 including an isolation structure around component A;

[0086] Fig. 22 shows a group and a component placed on a layout surface, where the group is sensitive to electrical disruption from the component;

[0087] Fig. 23 shows the layout of the group and the component shown in Fig. 22 including an isolation structure around the perimeter of the group;

[0088] Fig. 24 is a list of components, including for each component a placement constraint and a size, for placement in a layout surface;

[0089] Fig. 25 is a tree structure formed from the components and placement constraints shown in Fig. 24;

[0090] Figs. 26 and 27 are respective initial and final layouts of the components of Fig. 24 determined from searches of the tree structure shown in Fig. 25;

[0091] Fig. 28 is a variation of the tree structure shown in Fig. 25;

[0092] Figs. 29 and 30 are respective initial and final layouts of the components of the tree structure in Fig. 28 determined from searches thereof;

[0093] Fig. 31 is a variation of the tree structure shown in Fig. 28;

[0094] Figs. 32 and 33 are respective initial and final layouts of the components of the tree structure of Fig. 31 determined from searches thereof;

[0095] Fig. 34 is a variation of the tree structure shown in Fig. 31;

[0096] Figs. 35 and 36 are respective initial and final layouts of the components of the tree structure of Fig. 34 determined from searches thereof;

[0097] Fig. 37 is a tree structure having leaves that represent components and groups;

[0098] Figs. 38 and 39 are respective initial and final layouts of the components and group of the tree structure of Fig. 37 determined from searches thereof;

[0099] Fig. 40a shows the placements of components X and Y relative to a vertical symmetry line in the absence of an isolation structure;

[0100] Fig. 40b shows the placements of components X and Y relative to the vertical symmetry line in the presence of an isolation structure around component X;

[0101] Fig. 41 shows a variation of the tree structure shown in Fig. 25;

[0102] Fig. 42 shows the final placement of the components of the tree structure shown in Fig. 41 determined from searches thereof;

- [0103] Fig. 43 shows another variation of the tree structure shown in Fig. 25;
- [0104] Fig. 44 shows the final placement of the components of the tree structure shown in Fig. 43 determined from searches thereof;
- [0105] Fig. 45 shows another variation of the tree structure shown in Fig. 25;
- [0106] Fig. 46 shows the final placements of the components of the tree structure shown in Fig. 45 determined from searches thereof;
- [0107] Fig. 47 shows another variation of the tree structure shown in Fig. 25; and
- [0108] Fig. 48 shows the final placement of the components of the tree structure shown in Fig. 47 determined from searches thereof.

### DETAILED DESCRIPTION OF THE INVENTION

[0109] The present invention will be described with reference to the accompanying figures where like nomenclature refer to like elements. Moreover, the terms left, right, top, bottom, horizontal, vertical and the like are used with reference to the accompanying figures and are not to be construed as limiting the invention.

[0110] The present invention is embodied in computer readable program code that executes on a processor of a stand-alone or networked computer or workstation that includes a processor, computer storage, an input/output system, a media drive, such as a disk drive, CD ROM drive, etc., and a computer-useable storage medium capable of storing the computer software that embodies the present invention. Under the control of the computer readable program code, the processor is capable of configuring and operating the computer system in a manner to implement the present invention. Computer systems of the type described above are well known in the art and have not been described herein for purpose of simplicity.

[0111] With reference to Fig. 24, the computer system receives as input a plurality of components, e.g., A, B, Q, X and Y, with each component having a placement constraint and a size, e.g., height and width, associated therewith. Each placement constraint can designate the component associated therewith for either non-symmetric placement or symmetric placement. Moreover, each symmetric placement constraint can further designate the component associated therewith for pair-symmetric placement, self-

symmetric placement, asymmetric placement or fixed symmetric offset placement. Each of these placement constraints will be described in greater detail hereinafter.

**[0112]** With reference to Fig. 25 and with continuing reference to Fig. 24, once each component, its placement constraint and its size have been received, the computer readable program code causes the processor to create a tree structure 2 that includes a plurality of nodes 4-10 and a plurality of leaves 12-20. In tree structure 2, the plurality of nodes includes root node 4 and interior nodes 6, 8 and 10. The plurality of leaves includes leaf 12 associated with component B; leaf 14 associated with component A; leaf 16 associated with component Y; leaf 18 associated with component Q; and leaf 20 associated with component X.

**[0113]** With reference to Fig. 26 and with continuing reference to Figs. 24 and 25, root node 4 represents a global symmetry line 24 that subdivides a layout surface 54 into a first side 56 and a second side 58. Each interior node 6, 8 and 10 represents a slicing line 26, 28 and 30, respectively, that prescribes the relative positions of the components connected thereto on first side 56, second side 58, or both sides 56 and 58 of layout surface 54. The placement of global symmetry line 24 and the use of slicing lines 26-30 for placement of components on one or both sides 56 and 58 of layout surface 54 will be described in greater detail hereinafter.

**[0114]** The component associated with each leaf 12-20 is tagged for placement on first side 56, second side 58, or on both sides 56 and 58 of global symmetry line 24, and, hence, layout surface 54. In Fig. 25, the solid boxes around components B and A of leaves 12 and 14, respectively, indicate that components B and A are tagged for placement on the left side of global symmetry line 24; the dashed boxes around components Y and X of leaves 16 and 20, respectively, indicate that components Y and X are tagged for placement on the right side of global symmetry line 24; the solid box including cross hatching extending from the lower left to the upper right around component Q of leaf 18 indicates that component Q is tagged for pair-symmetric placement on both sides of global symmetry line 24. Each node 4-10 has a first branch 34-40 descending from a first side thereof and a second branch 44-50, respectively, descending from a second side thereof.

[0115] Desirably, the computer readable program code causes the processor to randomly create tree structure 2 and to randomly tag each component designated for non-symmetric placement on the first side or the second side of global symmetry line 24. To this end, the direction, i.e., horizontal or vertical, of global symmetry line 24 or any slicing line 26, 28 or 30, is randomly selected. In practice, when tree structure 2 is initially created, it is desirable that each interior node be connected to (i) two other interior nodes, (ii) an interior node and a leaf, (iii) or two leaves in the next lower level of tree structure 2. However, this is not to be construed as limiting the invention. It is also desirable that interior nodes 6-10 extend in one direction from root node 4 to form therewith an edge of tree structure 2. However, this is not to be construed as limiting the invention.

[0116] With reference to Fig. 26 and with continuing reference to Figs. 24 and 25, the use of tree structure 2 for placement of components A, B, Q, X and Y will now be described. Desirably, a structured search, for example, a depth-first search, is conducted of tree structure 2 to determine the initial placement of components. Specifically, starting at node 4, the search establishes global symmetry line 24 in a layout surface 54. Next, the search algorithm advances down the edge of tree structure 2, via first branches 34, 36 and 38, and interior nodes 6 and 8, until it reaches interior node 10 which has no interior nodes attached thereto deeper in tree structure 2. This first search of tree structure 2 can be utilized to initially place components on the left side, the right side, or simultaneously on the left and right sides of global symmetry line 24. For purpose of illustration, this first search of tree structure 2 will be described in connection with the placement of components on the left side of global symmetry line 24. However, this is not be construed as limiting the invention.

[0117] From interior node 10, the search algorithm advances to leaf 20 via first branch 40. Upon determining that component X associated with leaf 20 is tagged for placement on the right side of global symmetry line 24, the search algorithm backtracks to node 10 and advances to leaf 18 via second branch 50. Upon determining component Q associated with leaf 18 is tagged for pair-symmetric placement on both sides of global symmetry line 24, the search algorithm places a first instance of component Q on the left side of global symmetry line 24, with a right side 60 of this first instance of component Q

adjacent global symmetry line 24, and with a left side 62 placed the width of component Q away from right side 60. In Fig. 26, the width of each component placed in layout surface 54 is perpendicular to global symmetry line 24 while the height of each component is parallel to global symmetry line 24. However, this is not to be construed as limiting the invention since the height of each component can be positioned perpendicular to global symmetry line 24 and the width of each component can be positioned parallel to global symmetry line 24 if desired.

**[0118]** After placing the first instance of component Q, the search algorithm backtracks to interior node 8 and advances to leaf 16 via second branch 46. Upon discovering that component Y associated with leaf 16 is tagged for placement on the right side of global symmetry line 24, the search algorithm backtracks to interior node 6 and advances to leaf 14 via second branch 46.

**[0119]** Upon determining component A associated with leaf 14 is tagged for placement on the left side of global symmetry line 24, the search algorithm places component A on the left side of global symmetry line 24 relative to component Q based upon the orientation of slicing line 26 associated with interior node 6. More specifically, the horizontal orientation of slicing line 26 instructs the search algorithm that component A is to be placed above (or below) any component(s) placed thus far during this first search of tree structure 2.

**[0120]** The placement of component A above component Q in Fig. 26 can be based upon a rule that components connected to an interior node that represents a horizontal slicing line via the interior node's second branch are positioned above components connected to the interior node via its first branch. However, this is not to be construed as limiting the invention since this placement rule can be modified whereupon components connected to the interior node via its first branch are positioned above components connected to the interior node via its second branch. Alternatively, the decision to place component A above component Q, or vice versa, can be based upon the order of placement of component A and component Q by the search algorithm. For example, a rule can be established that components first placed by the search algorithm are placed below (or above) components that are placed thereafter. Desirably, however, once

established, the rule for placement of components connected to a node having a horizontal slicing or symmetry line associated therewith is fixed.

**[0121]** Next, search algorithm backtracks to root node 4 and advances to leaf 12 via second branch 44. Upon determining that component B associated with leaf 12 is tagged for placement on the left side of global symmetry line 24, the search algorithm places component B on the left side of global symmetry line 24 on the sides of components A and Q opposite global symmetry line 24.

**[0122]** The decision to place component B on the sides of components A and Q opposite global symmetry line 24 can be based on a rule that components connected to root node 4 via its first branch 34 are positioned closer to global symmetry line 24 than components connected to root node 4 via its second branch 44. However, this is not to be construed as limiting the invention since this placement rule can be modified whereupon components connected to root node 4 via its second branch 44 are positioned closer to global symmetry line than components connected to root node 4 via its first branch 34. Alternatively, a rule can be established that components first placed by the search algorithm are placed closer to global symmetry line 24 than components placed thereafter, or vice versa. However, once established, the rule for placement of components connected to a node having a vertical slicing or symmetry line associated therewith is fixed.

**[0123]** Next, another structured search of tree structure 2 can be conducted to determine the placement of components on the right side of global symmetry line. Initially, the search algorithm advances down the edge of tree structure 2 until it reaches interior node 10. Next, the search algorithm advances to leaf 20 via first branch 40. Upon determining that component X associated with leaf 20 is tagged for placement on the right side of global symmetry line 24, the search algorithm places component X on the right side of global symmetry line 24. Next, the search algorithm backtracks to node 10 and advances to leaf 18 via second branch 50. Upon determining that component Q associated with leaf 18 is tagged for placement on both sides of global symmetry line 24, the search algorithm places a second instance of component Q (hereinafter Q') on the right side of global symmetry line 24 on a side of component X opposite global symmetry line 24.

[0124] Next, the search algorithm backtracks to node 8 and advances to leaf 16 via second branch 48. Upon determining that component Y associated with leaf 16 is tagged for placement on the right side of global symmetry line 24, the search algorithm places component Y on the right side of global symmetry line 24 on a side of component Q' opposite component X.

[0125] After placing component Y, the search algorithm backtracks to node 6 and advances to leaf 14 via second branch 46. Upon learning that component A associated with leaf 14 is tagged for placement on the left side of global symmetry line 24, the search algorithm backtracks to node 4 and advances to leaf 12 via second branch 44. Upon determining that component B associated with leaf 12 is designated for placement on the left side of global symmetry line 24, the search algorithm terminates the second search of tree structure 2.

[0126] In the foregoing description, first and second searches of tree structure 2 were performed to place components on the respective left and right sides of global symmetry line 24. However, as noted above, a single structured search of tree structure 2 can be performed to initially place components on both sides of global symmetry line 24. Hence, the foregoing description of two structured searches to place components on the left and right sides of global symmetry line 24 is not to be construed as limiting the invention.

[0127] With reference to Fig. 27 and with continuing reference to Figs. 25 and 26, once the initial placement of components A, B, Q, Q', X and Y is complete, the final location of each component is determined by conducting another structured search of tree structure 2. Initially, the search algorithm advances along the edge of tree structure 2 until it reaches interior node 10. Next, the search algorithm advances to leaf 20 via first branch 40 where the search algorithm sets a left side 72 of component X adjacent global symmetry line 24 and a right side 74 of component X the width of component X away from left side 72. The distance between global symmetry line 24 and right side 76 of component X is stored in memory for subsequent use.

[0128] Next, the algorithm backtracks to interior node 10 and advances to leaf 18. Since component Q associated with leaf 18 is tagged for pair-symmetric placement, the search algorithm utilizes the position of right side 74 of component X to place a right side

60 of component Q and a left side 76 of component Q' the same distance from global symmetry line 24, with the left side 76 of component Q' adjacent right side 74 of component X. The algorithm then stores in memory for future use the distance that the left side 62 of component Q or the right side 78 of component Q' are from global symmetry line 24.

[0129] The algorithm then backtracks to interior node 6 and advances to leaf 14. Since component A has been placed above component Q previously, the algorithm can utilize the information stored in memory regarding the distance left side 62 or right side 78 of component Q or Q' is from global symmetry line 24 to place left side 66 of component A the same distance from global symmetry line 24. Since left side 66 and left side 62 of components A and Q, respectively, are in alignment the same distance from global symmetry line 24, it is not necessary to store in memory the distance between left side 66 and global symmetry line 24.

[0130] If, however, the width of component A were such that placement of the left side 62 of component Q and the left side 66 of component A in alignment resulted in component A overlapping global symmetry line 24, the search algorithm would place the right side 64 of component A adjacent global symmetry line 24 and the distance between left side 66 of component A and global symmetry line 24 would be stored in memory.

[0131] Next, the search algorithm backtracks to root node 4 and advances to leaf 12. Utilizing the greater of the distance left side 62 of component Q or left side 66 of component A is placed from global symmetry line 24, the search algorithm places right side 68 of component B adjacent left side 62 or left side 66 of components Q or A, respectively. Thereafter, since the final locations of components A, B, Q, Q', X and Y have been determined, the algorithm terminates horizontal placement of these components.

[0132] If desired, subject to maintaining the pair-symmetric placement of components Q and Q', components A, B, Q, Q', X and Y can be moved as necessary, either upwardly or downwardly parallel to global symmetry line 24, to minimize any vertical spacing therebetween. To this end, the topmost edge and bottommost edge of components A, B, Q, Q', X and Y can be utilized to establish top and bottom boundaries of layout surface 54 for the purpose of moving the components upwardly or downwardly. Similarly, left



side 70 of component B and right side 90 of component Y can be utilized to establish the leftmost and rightmost boundaries of layout surface 54.

**[0133]** In the foregoing description, horizontal slicing line 26 and vertical slicing lines 28 and 30 were utilized to place components on opposite sides of global symmetry line 24. If however, global symmetry line 24 was oriented horizontally, the positions of components A, B, Q, Q', X and Y relative to each other and said horizontal global symmetry line can be determined from tree structure 2 simply by replacing each horizontal slicing line with a vertical slicing line and vice versa. Thereafter, one or more searches of the modified tree structure can be conducted in order to determine the initial placement of components on opposite sides of the horizontal global symmetry line and another search of the modified search tree can be conducted to determine the final placement of the components on opposite sides of the horizontal global symmetry line.

**[0134]** With reference to Fig. 28, suppose a tree structure 2a is formed by inserting an interior node 92 between interior nodes 8 and 10 of tree structure 2. In tree structure 2a, first branch 38 can connect interior node 92 to interior node 8 while a first branch 94 can connect interior node 92 to interior node 10. A second branch 96 can connect a leaf 98 to interior node 92. A component S, tagged for self-symmetric placement on both sides of global symmetry line 24, can be associated with leaf 98 and a horizontal slicing line 102 can be associated with interior node 92. In addition, suppose, horizontal slicing line 26 in tree structure 2 is replaced with vertical slicing line 26a in tree structure 2a. Other than the foregoing, tree structure 2a in Fig. 28 is the same as tree structure 2 in Fig. 25.

**[0135]** With reference to Fig. 29 and with continuing reference to Fig. 28, the search algorithm can perform one or more structured searches of tree structure 2a to initially place components A, B, Q and  $\frac{1}{2}S$  on the left side of global symmetry line 24 and to place components Q',  $\frac{1}{2}S'$ , X and Y on the right side of global symmetry line 24. More specifically, during the one or more searches of tree structure 2a, when the search algorithm determines component S associated with leaf 98 is tagged for self-symmetric placement, one-half of component S is initially placed on the left side of global symmetry line 24 and the other half of component S is initially placed on the right side of global symmetry line 20 in the manner described above in connection with tree structure 2. The remaining components of tree structure 2a are initially placed in the manner described

above in connection with tree structure 2. The initial placement of the components of tree structure 2a is shown in Fig. 29.

[0136] Once the initial placement of components is complete, another search of tree structure 2a is conducted to determine the final location of each component in Fig. 29. The final placement of the components of tree structure 2a is accomplished in the manner described above in connection with the final placement of components associated with tree structure 2. The final placement of the components of tree structure 2a is shown in Fig. 30.

[0137] With reference to Fig. 31 and with continuing reference to Fig. 28, re-tagging component X from placement on the right of global symmetry line 24, as shown in Fig. 28 to placement asymmetrically on both sides of global symmetry line 24, as shown in Fig. 31, results in tree structure 2a being converted to tree structure 2b. The re-tagging of component X for asymmetric placement on both sides of global symmetry line 24 includes associating with component X data 104 that the search algorithm utilizes to determine what percentages of component X to place on both sides of global symmetry line 24 during the initial placement of component X. Specifically, as shown in Fig. 32, when the search algorithm determines that component X associated with leaf 20 is tagged for asymmetric placement on both sides of global symmetry line 24, the search algorithm utilizes data 104 to place one portion, e.g.,  $1/3$ , of component X on the left side of global symmetry line and to place the other portion, e.g.,  $2/3$ , of component X on the right side of global symmetry line 24.

[0138] The initial placement of the remaining components of tree structure 2b is then performed in the same manner as discussed above for the placement of components associated with tree structure 2. Once the initial placement of the components of tree structure 2b is complete, another search of tree structure 2b is performed to determine the final placement of the components of tree structure 2b shown in Fig. 33.

[0139] With reference to Fig. 34, re-tagging component X associated with leaf 20 in tree structure 2b from placement asymmetrically to placement a fixed symmetric offset on the right side of global symmetry line 24 changes tree structure 2b to tree structure 2c shown in Fig. 34. Performing one or more searches of tree structure 2c to initially place the components thereof results in the initial component placement shown in Fig. 35.

Performing another search of tree structure 2c produces the final placement of the components thereof shown in Fig. 36. As shown in Fig. 34, component X associated with leaf 20 is tagged for placement on the right side of global symmetry line 24. In addition, component X has associated therewith data 106 which includes a distance D component X is to be placed from global symmetry line 24. Thus, when performing the final placement of the components of tree structure 2c, the search algorithm places the left side of component X distance D from global symmetry line 24. If component X were tagged for placement on the left side of global symmetry line 24, the search algorithm would place component X on the left side of global symmetry line 24, with the right side of component X distance D from global symmetry line 24.

**[0140]** With reference to Fig. 37, additionally or alternatively, groupings of leaves and nodes can be placed on one or both sides of global symmetry line 24. To this end, one or more leaves, one or more local interior nodes and/or one or more local symmetry lines can be represented by a group which can be placed on one or both sides of global symmetry line 24 in the manner described above for components. For example, in Fig. 37, tree structure 112 includes root node 114, interior nodes 116-124, leaves 126-130, 136 and 138, and groups 132 and 134 connected in the illustrated manner.

**[0141]** In the foregoing description, two searches were performed of each search tree to determine the initial layout of components. However, as described above, a single search of the search tree can also be performed to determine the initial layout of components on both sides of global symmetry line 24. Hereinafter, the present invention will be described in connection with a performance of the single search of a search tree to determine the initial placement of components on either side of a symmetry line, either a global symmetry line or a local symmetry line.

**[0142]** With reference to Fig. 38 and with continuing reference to Fig. 37, utilizing tree structure 112, the search algorithm initially establishes a global symmetry line 164 associated with root node 114. Next, the search algorithm advances along the leftmost edge of tree structure 112 to interior node 116. Next, the search algorithm advances to leaf 126 associated with component B via first branch 142. In response to determining that component B is tagged for pair-symmetric placement, the search algorithm places copies B and B' of component B on the respective left and right sides of global symmetry

line 164. Thereafter, the search algorithm backtracks to interior node 116 and advances to interior node 118 via second branch 144. At interior node 118, the search algorithm advances to leaf 126 associated with component C via a first branch 146. Upon determining that component C is tagged for pair-symmetric placement, the search algorithm places copies C and C' of component C on the respective left and right sides of global symmetry line 164. Since copies C and C' of component C have already been placed, the search algorithm places copies C and C' above copies B and B'. However, this is not be construed as limiting the invention since other placement rules can be utilized.

[0143] Next, the search algorithm backtracks to interior node 118 and advances to leaf 130 associated with component D via second branch 148. Upon determining that component D is tagged for pair-symmetric placement, the search algorithm places copies D and D' of component D on the left and right sides of global symmetry line 164 above copies C and C', respectively.

[0144] Next, since the first and second branches of interior nodes 116 and 118 have already been searched, the search algorithm backtracks to root node 114. From there, the search algorithm advances down the rightmost edge of tree structure 112 to interior node 124. From there, the search algorithm advances to leaf 136 corresponding to component Z via first branch 156. Upon determining that component Z is tagged for placement on the right side of global symmetry line 164, the search algorithm places component Z to the right of components B', C' and D'.

[0145] Next, the search algorithm backtracks to node 124 and advances to leaf 138 corresponding to component X via second branch 158. Since component X is tagged for placement on the right side of global symmetry line 164 and since interior node 124 has vertical symmetry line 172 associated therewith, component X is placed on a side of component Z opposite global symmetry line 164.

[0146] With reference to Fig. 39 and with continuing reference to Figs. 37 and 38, next, the search algorithm backtracks to interior node 122 associated with horizontal symmetry line 174. From interior node 122, the search algorithm advances to leaf 134 representing a group B via first branch 160. Upon determining that group B is tagged for placement on the right side of global symmetry line 164, the search algorithm places a rectangle

representing group B, or the component(s) and any local symmetry line(s) associated with group B, above components Z and X. The placement of the rectangle representing group B above components Z and X can be based upon the order of placement of components Z and X and group B or the connection of group B to interior node 122 via first branch 160 and the connection of leaves 136 and 138 representing components Z and X, respectively, to interior node 122 via second branch 154. As shown in Fig. 37, group B includes a local tree structure 175 having a local root node 176 that represents a local, horizontal symmetry line 178, an interior node 180 and leaves 182 and 184 representing components E and F, respectively.

[0147] At a suitable time, the search algorithm performs a search of local tree structure 175 to determine the placement of components E and F in the rectangle representing group B. More specifically, commencing at local root node 176, the search algorithm establishes local, horizontal symmetry line 178. Thereafter, the search algorithm advances via interior node 180 and first branch 186 to leaf 182. Upon determining that component E associated with leaf 182 is tagged for asymmetric placement on both sides of local, horizontal symmetry line 178, the search algorithm places one-half of component E below horizontal symmetry line 178 and the other half of component E above horizontal symmetry line 178. Thereafter, the search algorithm backtracks to node 180 and advances to leaf 184 via second branch 188. Upon determining that component F is tagged for pair-symmetric placement on both sides of local, horizontal symmetry line 178, the search algorithm places copies F and F' of component F above and below local, horizontal symmetry line 178 on the side of component E opposite global symmetry line 164. The decision to place copies F and F' of component F to the right of component E can be based on a rule for placement of components connected to an interior node, e.g., local interior node 180, representing a vertical, local symmetry line 189 whereupon later placed components are placed further away from global symmetry line 164 than earlier placed components. However, this is not to be construed as limiting the invention.

[0148] After the search algorithm places a rectangle representing group B, or the component(s) and any local symmetry line(s) associated with group B, above components Z and X, the search algorithm backtracks from leaf 134 to interior node 120 via interior node 122 and second branch 152. From interior node 120, the search algorithm advances

to leaf 132. Upon determining that group A associated with leaf 132 is tagged for placement on the left side global symmetry line 164, the search algorithm places a rectangle representing group A, or the component(s) and any local symmetry line(s) associated with group A, on a side of components B, C and D opposite global symmetry line 164. As shown in Fig. 37, group A includes a local tree structure 189 having a local root node 190 that represents a local, horizontal symmetry line 192, an interior node 194 and leaves 196 and 198 representing a component A and a group C, respectively.

**[0149]** At a suitable time, the search algorithm performs a search of local tree structure 189 to determine the placement of components associated therewith in the rectangle representing group A. Specifically, commencing with local root node 190, the search algorithm initially places a local, vertical symmetry line 192 on a side of components B, C and D opposite global symmetry line 164 as shown in Fig. 39. Thereafter, the search algorithm advances to leaf 196 associated with component A via interior node 194 and first branch 200. Upon determining that component A is tagged for pair-symmetric placement on both sides of local, vertical symmetry line 192, the search algorithm places copies A and A' of component A on the respective left and right sides of local, vertical symmetry line 192.

**[0150]** Thereafter, the search algorithm backtracks to interior node 194 and advances to leaf 198 via second branch 202. Upon determining that group C associated with leaf 198 is tagged for pair-symmetric placement on both sides of local, vertical symmetry line 192, the search algorithm places copies of the rectangle representing group C on the left and right sides of local symmetry line 192 above copies A and A' of component A based on a rule for placement of components connected to a local, horizontal interior node, e.g., local interior node 194, representing a horizontal local symmetry line 203.

**[0151]** As shown in Fig. 37, group C includes local tree structure 204 having a local root node 206 that represents a local, vertical symmetry line 208, a leaf 210 representing a component Y and a leaf 212 representing a null. In practice, it is not necessary to include leaf 212 and second branch 216 connected thereto in local tree structure 204. However, second branch 216 and leaf 212 are included in group C shown in Fig. 37 for the purpose of illustration. At a suitable time, the search algorithm performs a search of local tree structure 204 associated with group C. Starting at local root node 206, the

search algorithm places copies 208 and 208' of local, vertical symmetry line 208 on the left and right sides of local symmetry line 192. Thereafter, the search algorithm advances to leaf 210 via first branch 214. Upon determining that component Y associated with leaf branch 210 is tagged for pair-symmetric placement on both sides of local symmetry line 208, the search algorithm places first copies Y and Y' of component Y on the left and right sides of local symmetry line 208 and places second copies Y and Y' of component Y on the left and right sides of local symmetry line 208'.

**[0152]** After the search algorithm places copies of the rectangle representing group C on the left and right sides of local symmetry line 92, or places copies of each component and each symmetry line associated with each copy of group C on the left and right sides of local symmetry line 192, the search algorithm backtracks to root node 114 via interior node 120 and second branch 150 and terminates the initial placement of the components associated with tree structure 112.

**[0153]** Once the initial placement of components associated with tree structure 112 is complete, another search of tree structure 112 can be performed in the manner discussed above in connection with tree structures 2, 2a, 2b and 2c, to determine the final placement of the components associated with tree structure 112 shown in Fig. 39.

**[0154]** In the foregoing description, the placement of each component with respect to a global or local symmetry line was based on the width of the component for a vertical symmetry line or the height of the component for a horizontal symmetry line. It is often, however, desirable to place an isolation structure on one or more sides of an object or a group in order to isolate it from outside disruptive sources, such as an electrical source, a thermal source, and the like, or to provide additional space to account for variances in manufacturing tolerances. To this end, the component or group associated with each leaf requiring an isolation structure can have associated therewith isolation data (not shown) that the search algorithm can utilize to increase the size of the component or the group to provide the necessary isolation structure on one or more sides thereof. For example, the isolation data can cause the search algorithm to include an isolation structure on the left side, right side, topside and/or bottom side of a component or group as desired. In a similar manner, an isolation structure can be associated with the global root node, a local root node, and/or an internal node. For nodes having isolation data associated therewith,

the size associated with each component connected to the node via its branches will be augmented by the isolation structure on one or more sides thereof.

**[0155]** With reference to Figs. 40a and 40b, once a component has been augmented with an isolation structure, the periphery of the isolation structure is utilized in place of the periphery of the component for purposes for placement of the component. For example, Fig. 40a shows the placement of a component X with its left edge adjacent a symmetry line 220 and with its right edge adjacent the left edge of a component Y. If, during placement of component X, the search algorithm determines that component X has isolation data associated therewith, the search algorithm adds isolation structure 222 around component X during placement and uses the periphery of isolation structure 222 in place of the periphery of component X for purposes of placement. Thus, after placement, the left and right edges of isolation structure 222 are positioned adjacent symmetry line 220 and the left edge of component Y, respectively. Thus, as can be seen, the left side of isolation structure 222 increases the distance between the left edge of component X and symmetry line 220 while the right side of isolation structure 222 increases the distance between the right edge of component X and the left edge of component Y.

**[0156]** The searches of a tree structure described above may not necessarily result in the most efficient placement of components. In order to automatically generate a new placement of components, the tree structure that resulted in the original placement of components can be amended or perturbed and searches of the thus perturbed tree structure can be performed to determine a new layout of components. For example, Fig. 27 shows the final layout of components A, B, Q, Q', X and Y resulting from tree structure 2 shown in Fig. 25. One perturbation of tree structure 2 can include changing the direction slicing line 26 from horizontal in Fig. 25 to vertical in the tree structure shown in Fig. 41. The final layout resulting from this change is shown in Fig. 42.

**[0157]** Another perturbation can include changing a component tagging whereupon the component placement changes from one side of a symmetry line to the other side of the symmetry line. For example, in Fig. 25, component X is tagged for placement on the right side of global symmetry line 24. If tree structure 2 is perturbed by tagging component X for placement on the left side of global symmetry line, as shown in Fig. 43,



the final layout of components shown in Fig. 44 results from the searches of the tree structure shown in Fig. 43.

[0158] Another perturbation of tree structure 2 in Fig. 25 can include swapping components, e.g., components Q and X to produce the tree structure shown in Fig. 45. After this swap, component Q tagged for pair-symmetric placement on both sides of global symmetry line 24 is associated with leaf 20 and component X tagged for placement on the right side of global symmetry line 24 is associated with leaf 18. The final placement of components resulting from the tree structure shown in Fig. 45 is shown in Fig. 46.

[0159] Another example of a tree perturbation includes swapping a node and a leaf. For example, swapping interior node 10 and leaf 14 in Fig. 25 results in the tree structure shown in Fig. 47. The final placement of components resulting from searches of the tree structure shown in Fig. 47 is shown in Fig. 48.

[0160] As can be seen from a comparison of the final layout shown in Fig. 27 to the final layout shown in each of Figs. 42, 44, 46 and 48, a simple perturbation of the tree structure shown in Fig. 25 produces dramatic changes in the layout of the components associated therewith. The four simple perturbations discussed above in connection with Figs. 41- 48 can be used either singly or in combination as desired to produce even more changes in the final layout of components. Another simple perturbation includes exchanging the width and the height of each of one or more components.

[0161] Moreover, the search algorithm can perform a plurality of tree perturbations, either stochastically and/or heuristically, in order to generate a plurality of different final layouts. More specifically, starting from an initial tree structure, the search algorithm can stochastically or heuristically perturb the original tree structure and then determine therefrom the final layout of components therefor. The search algorithm can then perturb the new tree structure to obtain another new tree structure from which the search algorithm can generate another final layout of components. This process can continue until a desired number of new tree structures and their associated final layouts have been produced.

[0162] The perturbation of tree structures can occur sequentially, that is, with each new tree structure being derived from the immediately preceding tree structure. However, the

search algorithm can be configured to return to a tree structure in the sequence other than the immediately preceding tree structure. Upon returning to this tree structure in the sequence, the search algorithm can perturb the tree structure in a manner differently than the previous perturbation thereof in order to generate another new tree structure that heretofore was not previously generated. From this newly generated tree structure, other new tree structures can then be generated. In this manner, the number of tree structures produced from an original tree structure can be greatly enlarged.

**[0163]** A “cost” can be determined for the final layout associated with each tree structure. This “cost” is a numerical value that quantifies the desirability or favorability of the corresponding final layout. The “cost” of each final layout can be determined in any suitable manner. One manner of determining the “cost” of a final layout includes forming a ratio of the total area of the components and, if provided, any isolation structures over the total area of a rectangle that bounds the components, and any isolation structures, associated with the final layout. As would be apparent to one of ordinary skill in the art, a cost determined in this manner having a value closer to 1 means that the components forming the final layout take up more space of the bounding rectangle than a final layout having a cost further away from the value 1.

**[0164]** Once a cost has been determined for the final layout associated with each tree structure, the layout having the most favorable cost associated therewith can be selected for implementation.

**[0165]** As can be seen, the present invention utilizes a tree structure configured to support all of the constraints necessary for placement of components and/or groups. The tree structure can be easily perturbed to generate new tree structures wherefrom new layouts that satisfy all the placement constraints can be generated. A cost associated with each layout can then be utilized to select an optimal layout for implementation.

**[0166]** The invention has been described with reference to the preferred embodiments. Obvious modifications and alterations will occur to others upon reading and understanding the preceding detailed description. It is intended that the invention be construed as including all such modifications and alterations insofar as they come within the scope of the appended claims or the equivalents thereof.